Semantic Reasoning based Video Database Systems

Duc A. Tran, Kien A. Hua, and Khanh Vu

School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816, USA.
{dtran, kienhua, khanh}@cs.ucf.edu

Abstract. A constraint of existing content-based video data models is that each modeled semantic description must be associated with time intervals *exactly within* which it happens and semantics not related to any time interval are not considered. Consequently, users are provided with limited query capabilities. This paper is aimed at developing a novel model with two innovations: (1) Semantic contents not having related time information can be modeled as ones that do; (2) Not only the temporal feature of semantic descriptions, but also the temporal relationships among themselves are components of the model. The query system is by means of reasoning on those relationships.
To support users' access, a video algebra and a video calculus as formal query languages, which are based on semantic relationship reasoning, are also presented.

1 Introduction

Increasingly, there are more and more demands for video to be managed by video database systems (VDBMSs), similar to the way alphanumerical data is managed in traditional databases. The most important issue confronting VDBMSs is the description of the structure of video data in a form appropriate for querying, updating, and presentation. To address this, there have been a number of proposals which can be grouped into two categories:

- **Physical feature based modeling** [13, 11, 10, 3, 4, 17]: Much research has been done in the area of video modeling/querying based on audio-visual features, such as audio, color, texture and motion, captured by image processing or computer vision techniques. Those detected features are used as keys for users' retrieval and querying. However, non-expert users might not want to choose color or motion parameters to form a query. Additionally, semantics of a video do not reside only in how the video is built physically, so video data retrieval based on semantic contents seems more natural and preferable to users.
- **Semantic content based modeling**: A crucial property of a video information system is how it handles semantics. Most existing approaches like

this, which are very sparse, deal with models of video objects associated with their semantic descriptions. Examples of this group are [7, 9, 6].

The second direction has been dominant in finding semantic foundations for representing and querying video information due to its flexibility and capability of going far beyond the physical characteristics of video data. Semantics based models are in turn classified into two main tracks, *segmentation-based* and *stratification-based* models. The former [3, 5, 8, 15] first segment the video stream into a set of temporally ordered shots and then build a multilevel abstraction upon them. The drawback of this approach, as pointed out in [14], is the lack of flexibility and the incapability of representing semantics residing in overlapped segments. In a contrary direction, stratification models [14, 9, 1, 16, 12] segment contextual information of the video instead of simply partitioning. The video units, each called a stratum, can overlap and encompass each other and are associated with a time interval corresponding to a physical segment in the video stream. Most recently is [7] which extends the conventional stratification concept by allowing an event to associate with multiple time intervals.

In overall, current approaches only focus on capturing a time interval, or a set of time intervals, associated to a given semantic description. Descriptions without any related time tag are not taken into account, hence the scope of questions about the video is very limited. Our work takes into account the importance of *semantics-semantics relationships*. It is also encouraged by the fact that different information extraction techniques bring out various types of knowledge. In the context of video data, such knowledge can be either time information of an event, or temporal relationships among events, or both. It is a good idea to take advantage of all knowledge sources as much as possible in order to best strengthen the VDBMS. We study a video data model, called SemVideo, with the following properties: (1) Some semantic contents not having related time information are modeled as ones that do; (2) Not only the temporal feature of semantic descriptions, but also the temporal relationships among themselves are components of the model.

Based on the model, we derive mechanisms for the query system that provides reasonably powerful capabilities to end-users for their efficiently questioning about the database. In particular, we propose two formal query languages which are a video algebra and a video calculus. They exhibit a comprehensive set of formations working at both video and within-video level.

The rest of this paper is organized as follows. In section 2, we formally present the details of SemVideo. The query languages are described in section 3. Finally, we give some concluding remarks in section 4.

2 Semantic Video Data Model

In this section, we attempt to address the issues from the aforementioned discussion. The new model is called the *Semantic Video Data Model* (SemVideo) because the primary foci are semantic descriptions, which are the meaningful information about the video, and their mutual relationships. We remove the constraints of previous content-based models and add a new dimension to be managed by the database, which is inter-description relationships.

Our SemVideo model has the following types of information: (1) Videos: The database manages many videos, each being represented by a unique identifier; (2) Video objects: Basically, a video object is nothing but a video sequence. In this paper, it is further extended to be a set of video segments that satisfy some constraint. Video objects are abstract and not really stored in the database; (3) Semantic objects: A semantic object is a description of knowledge about the video. It has a number of attributes, each having a corresponding value. Each semantic object in the video has a unique identifier to differentiate from others; (4) Entities: An entity can be either a video, a video object or a semantic object; (5) Relationships: A relationship is an association between two entities. It can be time related or semantic-related. Note that, in existing models, relationships are limited to only time relationships between video objects, based on which the relationships between semantic objects are established.

Let Ω be the set of all possible (time) **intervals** which can be written as $[t_1, t_2]$ where t_i's are integers and $t_1 \leq t_2$. Here comes the description of the proposed model.

Definition 1. *[Semantic Video Database] A* **semantic video database** *VDB is defined as a 7-tuple* $VDB = (\Sigma, \Delta, \Gamma, f_{DOM}, f_\Gamma, f_\Delta, f_{REL})$
where

- Σ*: Set of videos in the database. Each video is represented by a unique identifier.*
- Γ*: Set of semantic object attributes.* $TIME$ *is an element of this set regarding the time information for a semantic object.* $CONTAIN$ *is an attribute specifying what other semantic objects also happen during the current one happens.* ID *is a mandatory attribute of any semantic object.*
- f_{DOM} *is a function to map an attribute* γ *to its* value domain $f_{DOM}(\gamma)$. *Especially,* $f_{DOM}(TIME) = \Omega$, $f_{DOM}(CONTAIN) = 2^{\aleph}$ *and* $f_{DOM}(ID) = \aleph$*. In the other cases, a value domain can be* $2^{\aleph}$, $2^{\Re}$ *or* $2^{\Im}$ *where* $\aleph$ *is the set of natural numbers,* $\Re$ *the set of real numbers, and* $\Im$ *the set of strings.*
- Δ*: Set of all* possible semantic objects, *each being defined as the following tuple* $\delta = (ID\text{: soid}, \gamma_1\text{: } value_1, \gamma_2\text{: } value_2, ..., \gamma_n\text{: } value_n)$ *where soid is the identifier of the semantic object,* $\{\gamma_1, \gamma_2, .., \gamma_n\}$ *is a subset of* Γ *and* $value_i$ *is an element of* $f_{DOM}(\gamma_i)$ *for i from 1 to n.*
- f_Γ: $\Sigma \rightarrow 2^\Gamma$ *is a mapping function from* Σ *to the set of all subsets of* Γ*. For each video* σ *in* Σ, $f_\Gamma(\sigma)$ *gives a subset of* Γ*, that is the set of attributes used for the video.*
- f_Δ: $\Sigma \rightarrow 2^\Delta$*. For each video* σ *in* Σ, $f_\Delta(\sigma)$ *returns a set of tuples, each being a possible semantic object whose attribute set is a subset of* $f_\Gamma(\sigma)$*. Such tuples are classified as* semantic objects *of video* σ.
- f_{REL} *is a mapping function that, given a video* σ*, returns a time relation of the following form* $f_{REL}(\sigma)$: $f_\Delta(\sigma) \times f_\Delta(\sigma) \rightarrow ALLEN \cup \{\text{VOID}\}$ *where*

ALLEN is the set of the thirteen interval relations given in [2]. If the returned value is VOID, the two semantic objects have no time relationship. $f_{REL}(\sigma)$ *is called the* relationship function *for the video* σ.

Note that the SemVideo model encompass the (conventional or extended) stratification model. A distinguishing feature of SemVideo is the allowance of a semantic object not associated with a time interval to be captured and that the relationship among semantic objects (represented by relationship function) is a component of the model. Time information of a semantic object can be computed based on these relationships and already-known temporal information of other objects. Before we go any further, let us give an example of a video database where only one video is examined. The video is part of the movie *Assassins*. To know what this segment of movie is about, consider the following script:

"Tiled roofs, the stark white stucco of a colonial town square. Black iron bars at a bank. A briefcase carried in a man's hand. A sniper's rifle being assembled. Thick blocks of hundred dollar bills. Placed in the briefcase. A man's teeth as he smiles grimly at the sight. The briefcase snaps shut. A vault door slams. Rubber soles walk a tiled floor. Ahead, brilliant, white light suffuses the exit. We're either going outdoors or over to the other side. A long rifle silencer juts from a window. We see the shooter from behind, a view over his shoulder. In the bank, the man crushes out a cigarette. Only the plaza pigeons notice. As they take flight a man lies dead on the cobblestones and as we look up toward the window, there is nothing there. The pigeons wheel above the plaza. We follow, finally losing them to the sky. The sky changes from gray to blue."

Benefiting from knowledge obtained by several extraction techniques, we can build a video database as follows. In our video database, $\Sigma = \{\sigma\}$, $\Gamma = \{$ID, SUBJECT, ACTION, TIME$\}$, $f_\Gamma(\sigma) = \Gamma$, f_{DOM}(SUBJECT) = f_{DOM}(ACTION) = $2^{\Im}$, $f_\Delta(\sigma) = \{\delta_i\}$ for i = 1..15.

δ_1 = (1, SUBJECT: {roof, town square}, TIME: [0, 3])

δ_2 = (2, SUBJECT: {bank, black iron bars})

δ_3 = (3, ACTION: {carry}, SUBJECT: {briefcase, hand}, TIME: [4, ∞])

δ_4 = (4, ACTION: {assemble}, SUBJECT: {rifle, murderer}, TIME: [5, ∞])

δ_5 = (5, ACTION: {place}, SUBJECT: {money, briefcase}, TIME: [0, 7])

δ_6 = (6, ACTION: {smile}, SUBJECT: {teeth, murderer})

δ_7 = (7, ACTION: {shut}, SUBJECT: {briefcase}, TIME: [7, ∞])

δ_8 = (8, ACTION: {slam}, SUBJECT: {vault door})

δ_9 = (9, ACTION: {walk}, SUBJECT: {soles}, TIME: [0, 8])

δ_{10} = (10, SUBJECT: {exit sign})

δ_{11} = (11, ACTION: {jut}, SUBJECT: {murderer, shooter, long rifle, window, shoulder}, TIME: [9, 12], CONTAIN: {12})

δ_{12} = (12, ACTION: {crush}, SUBJECT: {person, cigarette}, TIME: [8, 10] $\sqcup$ [12, 13])

δ_{13} = (13, ACTION: {die}, SUBJECT: {person})

δ_{14} = (14, ACTION: {fly}, SUBJECT: {pigeon}, TIME: [14, 15])

δ_{15} = (15, ACTION: {change color}, SUBJECT: {sky})

$f_{REL}(\sigma)(\delta_1, \delta_2)$ = BEFORE (i.e., δ_1 ends before δ_2 starts)

$f_{REL}(\sigma)(\delta_2, \delta_3)$ = DURING (i.e., they happen during each other)
$f_{REL}(\sigma)(\delta_3, \delta_5)$ = BEFORE, $f_{REL}(\sigma)(\delta_4, \delta_5)$ = DURING
$f_{REL}(\sigma)(\delta_4, \delta_{10})$ = BEFORE, $f_{REL}(\sigma)(\delta_7, \delta_8)$ = BEFORE
$f_{REL}(\sigma)(\delta_8, \delta_9)$ = BEFORE, $f_{REL}(\sigma)(\delta_{11}, \delta_{13})$ = BEFORE
$f_{REL}(\sigma)(\delta_{13}, \delta_{14})$ = IN, $f_{REL}(\sigma)(\delta_{14}, \delta_{15})$ = BEFORE

Note that semantic objects can have TIME information or not and their relations are represented using the relationship function f_{REL} as above.

3 Query Languages

We now come to introducing content-based query mechanisms to support users' access. Given a number of videos, semantics about them are stored in the database using the model. Users can use them to retrieve various information of interest. To set the stage for a detailed look at video queries and query languages, we begin by clarifying what kinds of outputs are allowed in the querying system. We focus on answers of the following types: (1) a Boolean value (2) a video object (3) a semantic object (4) a video.

We introduce two formal query languages associated with SemVideo model, they are *video algebra* and *video calculus*. Prior to that, we need to define the operators to compare among attribute values of semantic objects, and then define what a selection condition is, which is useful for language descriptions later.

Given a semantic object δ, let $\gamma(\delta)$ be the value of attribute γ of δ.

Definition 2. *[Interval Operators]* **I-operators** *(***I** *stands for "interval") are operators applied on intervals. They ($\sqsubseteq$, $\sqsupseteq$, $\nearrow$, $\nwarrow$) are introduced as follows, where p and q are intervals: (1) $p \sqsubseteq q$: any semantic property of p must be true for q (2) $p \sqsupseteq q$: $p \sqsupseteq q$ iff $q \sqsubseteq p$ (3) $p \nearrow q$: the video segment according to p is before that according to q (4) $p \nwarrow q$: $p \nwarrow q$ iff $q \nearrow p$.*

Definition 3. *[Attribute Operators] Let $S \in \{\aleph, \Re, \Im\}$. And suppose that $v \in 2^S$, and $c \in S$.* **Attribute operators** *$\prec$, $\succ$, $\sim$ are defined as follow: (1) $S \neq \Im$: $v \prec c$ iff there exists an element $c' \in v$ such that $c' < c$. (2) $S \neq \Im$: $v \succ c$ iff there exists an element $c' \in v$ such that $c' > c$. (3) $v \sim c$ iff $c \in v$. I-operators are the special attribute operators which are applied on Ω values.*

Definition 4. *[Atomic Selection Condition] An* **atomic selection condition** *is defined as follows: (1) $a = b$ and $a \neq b$ are atomic selection conditions if a and b (variables or constants) are of the same type. (2) $a < b$, $a > b$, $a \leq b$ and $a \geq b$ are atomic seletion conditions if a and b are of the same type $\mathcal{T}$ and $\mathcal{T}$ has orderings $<$, $>$, $\leq$ and $\geq$ on its instances.*

Definition 5. *[Selection Condition] A* **selection condition** *is a boolean combination (i.e., an expression using the logical connectives $\neg$, $\vee$ and $\wedge$) of terms that have one of the following forms: (1) an* atomic-condition *(2) γ* **op** *$constant_1$ (3) $f_{REL}(\sigma_1)(\delta_1, \delta_2) = constant_2$ (4) $f_{REL}(\sigma_1)(\delta_1, \delta_2) = f_{REL}(\sigma_2)(\delta_3, \delta_4)$; where* atomic-condition *is an atomic selection condition, σ_i a video,* **op** *an attribute* *operator in* $\{\sqsubseteq, \sqsupseteq, \nearrow, \nwarrow, \prec, \succ, \sim\}$, γ *a semantic attribute,* δ_i *an semantic object, constant*$_i$ *a constant value so that* $\{constant_1\} \in f_{DOM}(\gamma)$ *and constant*$_2$ $\in ALLEN$.

The two query languages are described below.

3.1 Video algebra

Queries in algebra are composed using a collection of operators. According to the different types of outputs, operators are presented as follows.

- **Boolean operator** (Syntax: ?(*expr*), Return: a Boolean value): *expr* is a selection condition. If the condition is true, the answer is YES. Otherwise, the answer is NO.
- **Physical operation** (Syntax: $\phi_{expr}(\sigma)$, Return: an Ω value): *expr* is a selection condition, and σ is a video. The returned is an Ω value ω so that *expr* is true during ω.
- **Semantic operation** (Syntax: $\psi_{expr}(\sigma)$, Return: a value in Δ): *expr* is a selection condition, and σ is a video. The returned is a semantic object δ that is true about *expr* in the context of video σ.
- **Projection** (Syntax: $\pi_{expr}(\sigma)$, Return: a Σ value): *expr* is a selection condition, and σ is a video. The returned is a new video σ' that is part of σ. Those segments that do not satisfy *expr* are cut off. Related time information is modified to be consistent with the context of the new video.
- **Composition** (Syntax: $\chi(\sigma_1, \sigma_2)$, Return: a Σ value): σ_1 and σ_2 are videos. The returned is a new video σ that includes σ_1 and σ_2.
- **Updates**: In the real-world it is likely that the database will change over time. Update operations are α (Insertion), θ (Deletion) and μ (Modification).
  - `Insertion` (Syntax: $\alpha_{expr,expr'}(\sigma)$, Return: a Σ value): σ is a video, *expr* is a selection condition and *expr*$'$ is of the form $\gamma = constant$ where $\gamma \in f_\Gamma(\sigma)$, $\gamma \neq ID$ and $constant \in f_{DOM}(\sigma)(\gamma)$. First, all the semantic objects satifying *expr* are selected. If nothing is selected, a new semantic object equivalent to *expr* is inserted to the database.Otherwise, for each δ among them, if γ is an attribute of δ, its value will be changed to $\gamma(\delta) \sqcap constant$ if $\gamma = TIME$, $\gamma(\delta) \cup constant$ if $\gamma \neq TIME$. If γ is not an attribute of δ, it becomes an attribute with value *constant*.
  - `Deletion` (Syntax: $\theta_{expr,expr'}(\sigma)$, Return: a Σ value): σ is a video, *expr* is a selection condition and *expr*$'$ is of the form $\gamma = constant$ where $\gamma \in f_\Gamma(\sigma)$and $constant \in f_{DOM}(\sigma)(\gamma)$. Firstly, all the semantic objects satifying *expr* are selected. For each δ among them, if γ is not an attribute of δ, nothing is changed. Otherwise, if $\gamma = ID$ and $\gamma(\delta) = constant$, δ is deleted from the database. On the other hand, its value will be changed to $\gamma(\delta) \setminus constant$ if $\gamma \neq TIME$, ω if $\gamma = TIME$, where $\omega \sqsubseteq \gamma(\delta)$ and ω do not overlap with the interval *constant*.
  - `Modification` (Syntax: $\mu_{expr,expr'}(\sigma)$, Return: a Σ value): σ is a video, *expr* is a selection condition, and *expr*$'$ is of the form $\gamma = constant$

where $\gamma \in f_{\Gamma}(\sigma)$ and *constant* $\in f_{DOM}(\sigma)(\gamma)$. Firstly, all the semantic objects satifying *expr* are selected. For each δ among them, if γ is not an attribute of δ, nothing is changed. Otherwise, its value will be set to *constant*.

Examples of video algebra queries We now present several examples to illustrate how to write queries in video algebra. The database in the previous section is used for our examples. We will use parentheses as needed to make our algebra expressions unambiguous. Let σ represent the video. Let $\mathcal{A}$, $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{T}$ denote attributes ACTION, SUBJECT, CONTAIN, and TIME respectively.

The user is interested in the scene where a dead body appears after somebody has assembled a rifle, and wants to know if it is true that the scene belongs to interval [12, 15], he or she can take the following steps: (1) $\delta_1 = \psi_{(\mathcal{A}\sim\text{``}die\text{''})\wedge(\mathcal{S}\sim\text{``}person\text{''})}(\sigma)$ (2) $\delta_2 = \psi_{(\mathcal{A}\sim\text{``}assemble\text{''})\wedge(\mathcal{S}\sim\text{``}rifle\text{''})\wedge(\mathcal{S}\sim\text{``}murderer\text{''})}(\sigma)$ (3) $p = \phi_{f_{REL}(\sigma)(\delta_1,\delta_2)=BEFORE}(\sigma)$ (4) $?(p \sqsubseteq [12,15])$. The returned value of the last expression is the answer to the above query.

Now suppose that there is a collection of videos. The user might want to create a new video containing only segments of interest from the collection to be used later. For instance, the user wants a new video that is related to the murder only. He or she can write: (1) $\sigma'_1 = \pi_{\mathcal{S}\sim\text{``}murderer\text{''}}(\sigma_1)$ (2) $\sigma'_2 = \pi_{\mathcal{S}\sim\text{``}murderer\text{''}}(\sigma_2)$ (3) $\sigma' = \chi(\sigma'_1, \sigma'_2)$. Sometimes it is possible that the database contains incomplete information, so we may later want to modify or insert more knowledge. For instance, from some source of information extraction, we know that the scene in which the murderer smiles and his teeth are seen (semantic object δ_6) corresponds to period [6, 7], this information can be inserted to the database by the query: $\sigma = \alpha_{ID=6,\mathcal{T}=[6,7]}(\sigma)$

For some reason, semantic object δ_{12} does not exist anymore because of an earlier deletion. It is expected to take it out of semantic object δ_{11}. The query for this is $\sigma = \theta_{ID=11,\mathcal{C}=12}(\sigma)$

3.2 Video calculus

We introduce a video calculus, which can be considered an extension to the relational calculus, as an alternative to video algebra. It allows us to describe the set of answers without being explicit about how they should be computed. As in relational calculus, the language for writing *formulas* is the heart of our calculus. We now define these concepts formally, beginning with the notion of a formula.

Syntax of calculus queries

Definition 6. *[Atomic formula] Let Δ' be any set of semantic objects, σ_1, σ_2 video variables, ω_1 and ω_2 interval variables, δ_1 and δ_2 semantic object variables, and γ a semantic attribute. Let* **op** *denote an operator in the set $\{<, >, =, \leq, \geq, \neq, \sqsubseteq, \sqsupseteq, \nearrow, \nwarrow, \prec, \succ, \sim\}$. An* **atomic formula** *is one of the following: (1)*

*$\delta_1 \in \Delta'$ (2) ω_1 **op** ω_2 (3) $\gamma(\delta_1)$ **op** $\gamma(\delta_2)$ (4) $\gamma(\delta_1)$ **op** constant (5) $TIME(\delta_1)$ **op** ω (6) $f_{REL}(\sigma)(\delta_1,\ \delta_2) = constant$ (7) $f_{\Delta}(\sigma_1) = constant$ (8) $f_{\Delta}(\sigma_1) = f_{\Delta}(\sigma_2)$.*

Definition 7. *[Formula] A* **formula** *is recursively defined to be one of the following, where p and q are themselves formulas, and $p(T)$ denotes a formula in which the variable T appears: (1) any atomic formula (2) $\neg p$, $p \wedge q$, $p \vee q$, or $p \Rightarrow q$ (3) $\exists T(p(T))$, where T is a variable and $T \in \Sigma \cup \Omega \cup \Delta$ (4) $\forall T(p(T))$, where T is a variable and $T \in \Sigma \cup \Omega \cup \Delta$*

In the last two clauses above, $\exists$ and $\forall$ are two quantifiers in traditional logic, and are said to *bind* the variable T.

Definition 8. *[Free variable] A variable is said to be* **free** *in a formula or a sub-formula (a formula contained in a larger formula) if the (sub-)formula does not contain an occurrence of a quantifier that binds it.*

And now is time for the formal definition of a video calculus query.

Definition 9. *[Calculus query] A* **calculus query** *is defined as an expression of the form $\langle type-of-output\rangle\{T \mid p(T)\}$ or the form $\langle\gamma_1, \gamma_2, .., \gamma_n\rangle\langle type-of-output\rangle\{T \mid p(T)\}$ where γ_i for $i \in \{1, 2, .., n\}$ is a semantic attribute, T is the only free variable in the formula $p(T)$ and is of type $type-of-output \in \{\Sigma, \Omega, \Delta\}$.*

Semantics of calculus queries The answer to a calculus query $[\langle\gamma_1, \gamma_2, .., \gamma_n\rangle]$ $\langle$*type-of-output*$\rangle\{T \mid p(T)\}$, as we noted earlier, is the set of all values t of type *type-of-output* so that the formula $p(T)$ evaluates to TRUE with variable T assigned the value t. To complete this definition, we must state which value assignments to free variables in a formula make the formula TRUE.

A query is evaluated on a given instance of the video database. Let each free variable in a formula F be bound to a value. For the given assignments of values to variables, with respect to the given video database instance, the formula F is TRUE if one of the following holds:

- F is an atomic formula $\delta \in \Delta'$, and δ is a variable assigned a semantic object in the instance of Δ'.
- F is of an atomic formula $\omega \in \Omega$, and ω is a variable assigned an interval in the instance of Ω.
- F is an atomic formula $\gamma(\delta_1)$ **op** $\gamma(\delta_2)$, and the semantic objects assigned to δ_1 and δ_2 have attribute values $\gamma(\delta_1)$ and $\gamma(\delta_2)$ that make the comparison TRUE.
- F is an atomic formula $\gamma(\delta)$ **op** *constant*, and the semantic object assigned to δ has an attribute value $\gamma(\delta)$ equal to *constant*.
- F is an atomic formula $TIME(\delta)$ **op** ω, and the semantic object assigned to δ has an attribute value $TIME(\delta)$ that makes the comparison TRUE.
- F is an atomic formula $f_{REL}(\sigma)(\delta_1, \delta_2) = constant$, and the semantic objects assigned to δ_1 and δ_2 have value $f_{REL}(\sigma)(\delta_1, \delta_2)$ that makes the comparison TRUE.

- F is an atomic formula $f_\Delta(\sigma) = constant$, and the video assigned to σ has the set of semantic objects $f_\Delta(\sigma)$ equal to $constant$.
- F is an atomic formula $f_\Delta(\sigma_1) = f_\Delta(\sigma_2)$, and the videos assigned to σ_1 and σ_2 have the same set of semantic objects.
- F is of the form $\neg p$, and p is not TRUE; or of the form $p \wedge q$, and both p and q are TRUE; or of the form $p \vee q$, and one of them is TRUE; or of the form $p \Rightarrow q$, and q is TRUE whenever p is TRUE.
- F is of the form $\exists T(p(T))$, and there is some assignment of values to the free variables in $p(T)$, including T, that make it TRUE.
- F is of the form $\forall T(p(T))$, and there is some assignment of values to the free variables in $p(T)$ that make it TRUE no matter what value is assigned to variable T.

Examples of video calculus queries We now illustrate the video calculus through several examples. The video database example in section 2 is used for this purpose. The query "Show me the segments of video where the murderer appears with the gun" can be expressed by the calculus query: $\langle \Omega \rangle \{\omega \mid \forall \delta(\delta \in \Delta \wedge \mathcal{S}(\delta) \sim \text{"}murderer\text{"} \wedge \mathcal{S}(\delta) \sim \text{"}gun\text{"} \wedge \mathcal{T}(\delta) \sqsubseteq \omega\}$. If the user would like to find what the murderer does before committing murder while a man is placing money into a briefcase, the query below can be used:

$$\begin{aligned}\langle \mathcal{S}, \mathcal{A}\rangle \{\Delta\}\{\delta \mid \delta \in \langle \mathcal{S}, \mathcal{A}\rangle \{\Delta\}\{\delta \mid \mathcal{S}(\delta) \sim \text{"}murderer\text{"}\\ \wedge \exists \delta'(\delta' \in \Delta \wedge \mathcal{S}(\delta') \sim \text{"}person\text{"} \wedge \mathcal{A}(\delta') \sim \text{"}die\text{"} \wedge \mathcal{T}(\delta') \nwarrow \mathcal{T}(\delta)\}\\ \wedge \exists \delta'(\delta' \in \Delta \wedge \mathcal{S}(\delta') \sim \text{"}person\text{"} \wedge \mathcal{S}(\delta') \sim \text{"}money\text{"}\\ \wedge \mathcal{S}(\delta') \sim \text{"}briefcase\text{"} \wedge \mathcal{A}(\delta') \sim \text{"}place\text{"}\\ \wedge f_{REL}(\sigma)(\delta, \delta') = DURING)\}\end{aligned} \qquad (1)$$

The user might want a new video containing information about the murderer only, he or she can use the query: $\langle \Sigma \rangle \{\sigma \mid \forall \delta(\delta \in f_\Delta(\sigma) \wedge \mathcal{S}(\delta) \sim \text{"}murderer\text{"})\}$.

4 Concluding Remarks

In this paper, we introduced a semantic video data model called SemVideo with the following properties: (1) Some semantic contents not having related time information are modeled like ones that do; (2) Not only the temporal feature of semantic descriptions, but also the temporal relationships among themselves are components of the model.

An advantage of SemVideo model is that various types of knowledge captured by different semantics extraction techniques are utilized. In a long run, we expect that the metadata of a video database system will be built from various sources of information and continue being updated with more kinds of knowledge captured by more sources. The idea of SemVideo is on this track. Based on the model, we derived formalisms for the query system which provide reasonably powerful capabilities to end-users for their efficient questioning about the database. In particular, we proposed two formal query languages that are a (procedural) video algebra and a (declarative) video calculus and include a comprehensive set of query formations working at both video and within-video level.

References

1. S. Adali, K. S. Candan, S.-S. Chen, K. Erol, and V. S. Subrahmanian. Advanced video information system: Data structures and query processing. *ACM-Springer Multimedia Systems Journal*, 1996.
2. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM, Springer-Verlag*, 26(11), 1983.
3. E. Ardizzone and M. L. Cascia. Automatic video database indexing and retrieval. *Multimedia Tools and Applications*, 4(1), 1997.
4. S.-F. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong. Videoq: An automated content based video search system using visual cues. In *ACM Multimedia*, November 1997.
5. T.-S. Chua and L.-Q. Ruan. A video retrieval and sequencing system. *ACM Transactions on Information Systems*, 13(4), 1995.
6. Y. Day, S. Dagtag, M. Iino, A. Khoakhar, and A. Ghafoor. A multi-level abstraction and modeling in video databases. *ACM Springer-Verlag Multimedia Systems*, 7(5), 1999.
7. C. Decleir and M.-S. Hacid. A database approach for modeling and querying video data. In *IEEE Data Engineering*, Australia, 1999.
8. A. Hampapur, R. Jain, and T. Weymouth. Production model based digital video segmentation. *Journal of Multimedia Tools and Applications*, 1(1), 1995.
9. H. Jiang, D. Montesi, and K. Elmagarmid. Videotext database systems. In *IEEE Int'l Conf. on Multimedia Computing and Systems*, Ontario, Canada, June 1997.
10. J. Oh and K. A. Hua. An efficient and cost-effective technique for browsing, querying and indexing large video databases. In *ACM SIGMOD*, Dallas, TX, May 2000.
11. J. Oh, K. A. Hua, and N. Liang. A content-based scene change detection and classification technique using background tracking. In *SPIE Conf. on Multimedia Computing and Networking*, San Jose, CA, January 2000.
12. E. Oomoto and K. Tanaka. Ovid: Design and implementation of a video-object database system. *IEEE Trans. on Knowledge and Data Engineering*, 5, August 1993.
13. Y. Rui, S. Huang, and S. Mehrotra. Constructing table-of-content for videos. *ACM Springer-Verlag Multimedia Systems*, 7(5), 1999.
14. T. G. A. Smith and G. Davenport. The stratification system: A design environment for random access video. In *Proceedings of the 3rd Int'l Workshop on Network and Operating System Support for Digital Audio and Video*, La Jolla, CA, 1992.
15. D. Swanberg, C.-F. Shu, and R. Jain. Knowledge guided parsing in video databases. In *SPIE Conf. on Image and Video Processing*, volume 1908, San Jose, CA, February 1993.
16. R. Weiss, A. Duda, and D. Gifford. Content-based access to algebraic video. In *IEEE Int'l Conf. on Multimedia Computing and Systems*, Boston, USA, 1994.
17. A. Yoshitaka, Y. Hosoda, M. Yoshimisu, M. Hirakawa, and T. Ichikawa. Violone: Video retrieval by motion example. *Visual Languages and Computing*, 7, 1996.